Improving Multi-Agent Based Resource Coordination in Peer-to-Peer Networks

António L. Lopes and Luís M. Botelho
We, the Body and the Mind Research Lab of Adetti-ISCTE, Lisbon, Portugal
Email: {antonio.luis, luis.botelho}@iscte.pt

Abstract—**The distributed nature of peer-to-peer networks offers a solid ground for the deployment of environments where multiple agents, managing several resources, can cooperate in pursuing common and individual goals while achieving good overall performance. In this article we present a survey of recent work on the integration of multi-agent systems and peer-to-peer computing for resource coordination (including discovery, composition and execution of resources) and we propose an approach for optimizing resource coordination through the use of efficient peer-to-peer search mechanisms relying upon a powerful semantic overlay network. We also present an approach for the dynamic development of the required semantic overlay network from a network of randomly-connected peers.**

Index Terms—**multi agent systems, peer-to-peer networks, resource coordination, semantic overlay network**

I. INTRODUCTION

Peer-to-peer (*P2P*) networks have been the target of strong controversies, especially due to legal issues surrounding well-known applications such as *Napster* [1] and *Gnutella* [2] and the use of these applications in illegal file sharing of copyrighted material. However, the reason for the success of *P2P*-based systems does not lie on its potential to perform illegal actions, but rather on its distributed nature that treats each peer as equal and allows all peers to freely exchange resources without suffering from the threats of centralized systems, such as central point of failure, high maintenance costs and low scalability.

P2P networks are distributed environments in which peers can seamlessly exchange and share resources between them. In the last years, the challenge of *P2P* computing research has been to design and implement a robust, fault-tolerant and scalable distributed system of peers. The work on this area has first gained a spotlight when systems such as *Napster* and *Gnutella* have emerged as killer-applications for file-sharing between users across the world. On one side, we have the centralized approach of *Napster*, where peers use a unique central server to search for files and then use the *P2P* infrastructure to exchange them. On the other side, we have the totally distributed approach of *Gnutella*, where peers broadcast their file requests to the entire network through connections with neighbouring peers.

Both approaches suffer from scalability and robustness problems. *Napster* presents a central point of failure which compromises the overall stability of the system, whereas in *Gnutella*, the *flooding* mechanism used to spread users' requests becomes prohibitive when a large number of nodes exist in the system. These two systems (and their limitations) represent two sides of *P2P* networks and, since then, they have been used as a basis for improvement in almost all *P2P* computing research.

The evolution of the research in this area has delivered promising results (see section II for a survey), paving the way for developing more robust and scalable applications on top of *P2P* networks. Nonetheless, even though *P2P* computing presents some interesting properties that would enable creating high performance applications, it still lacks a degree of proactivity, which would enable higher autonomy, rationality and fairness [3]. Research on *P2P* computing has mainly addressed the efficient management of the network, treating each peer as a simple reactive node, with little or no autonomy at all, thus ignoring the potential for developing collaborative environments.

The research on intelligent agents has devoted considerable effort not only to communication and coordination, but also to reasoning, learning, and adaptive capabilities of each agent, seeking to increase their autonomy. However, multi-agent systems (*MAS*) often suffer from incapability to reorganize themselves in dynamic environments where no structure is present.

The combination of the distributed capabilities of *P2P* networks with the intelligence of autonomous agents appears to be very promising since it will allow the transparent access to large-scale distributed resources while maintaining high-availability, fault tolerance and low maintenance application deployment through self-organization [4]. We envision the use of both these technologies to create an *intelligent peer-to-peer* infrastructure [5] that will allow for a dynamic network of intelligent agents, while managing several resources, to cooperate in order to efficiently discover, compose and execute computational resources.

In previously published work [6], we have presented a generic approach for developing an innovative process for resource coordination in unstructured distributed environments, by combining these two technologies. In this article, we present a survey of recent research on *P2P*

computing and multi-agent systems for resource coordination and further extend the proposal described in [6] by presenting a concrete approach for optimizing resource coordination through the use of efficient *P2P* search mechanisms that rely upon a dynamically self-created powerful *semantic overlay network*.

The rest of the paper is organized as follows: section II presents a background on the existing search mechanisms in *P2P* networks; section III analyses recent research work on multi-agent based resource coordination; section IV describes our extended proposal; section V presents an evaluation of our preliminary simulations; and in section VI, we present some conclusions and guidelines for future work.

II. SEARCH MECHANISMS IN P2P NETWORKS

The research on *P2P* computing, which has been fuelled by the need to find more robust and scalable solutions, has been classifying *P2P* systems according to two traversal dimensions: *network structure* and *search mechanisms*. *Network structure* refers to the existence of some sort of structure according to which, some peers have different responsibilities or are hierarchically organized within the network. In terms of *network structure*, *P2P* systems can either be *pure* (also referred to as *unstructured*), where all peers are equal in responsibilities and no hierarchy exists; or *hybrid* (also referred to as *structured*), where peers are organized in specific hierarchies or some peers – also referred to as *super-peers* or *ultra-peers* – have different responsibilities, or peers are connected according to a specific structure based on the resources that they manage. The *search mechanism* dimension classifies *P2P* systems according to the way peers search other peers or specific resources in the network. According to this classification, *P2P* systems can employ *uninformed searches* (also referred to as *blind searches*), where each peer searches the network by randomly querying other peers; or *informed searches*, where a peer uses additional information about other peers' resources to select the peers that will be contacted during the search process [7]. In the following sub-sections we provide an overview of the existing search mechanisms both in unstructured and structured networks.

A. Search Mechanisms in Unstructured Networks

Unstructured *P2P* networks can be defined as networks where peers are randomly connected with other peers and no additional information is used to characterize those connections. In such networks, where peers cannot rely on any information to optimize the search process, searching a certain network resource or peer is often carried out by a *flooding* algorithm or a *random walk* algorithm. In the *flooding* algorithm (also referred to as *breadth-first search*), a peer broadcasts the search *query* to all of its neighbours, which in turn will apply the same process until the search *result* is found or some condition holds. A *Time-To-Live* constant is often used to stop the *flooding* propagation at a certain level. In the *random walk* algorithm (also referred to as *k-random* or

depth-first search – when $k = 1$), a peer chooses a k number of random neighbours to propagate the search *query*. These, in turn, will use the same process until the search *result* is found. Both algorithms present some disadvantages. *Flooding* increases network load with copies of the query message but may retrieve the results faster, whereas *random walk* reduces the network load but increases the search latency.

In recent years, some approaches designed search mechanisms that were based on some variations of these two algorithms. *Iterative deepening* [8] is an example of an effort to improve the use of *flooding* techniques. A peer, employing this search mechanism, initiates multiple *breadth-first* searches, over the iterations of the technique, with successively larger depth limits, until either the query is satisfied, or a maximum depth limit has been reached. To avoid having nodes processing the same request multiple times, *Resend* messages are used to guarantee that only nodes beyond the previous depth limit process the request; nodes within the previous depth limit only forward the request.

In an attempt to improve the effectiveness of search mechanisms in *P2P* networks, *informed searches* were introduced, offering the possibility to improve the performance of the discovery process by using information on peers and their resources. This information is obtained from previous queries. Knowing exactly which peers to use when propagating a *query* can help reduce the network load (less *flooding*) while improving the search performance.

Routing Indices [9] allow nodes to forward queries to a subset of neighbours that are the best candidates to satisfy the query. The subset of candidate neighbours is identified by evaluating an index table that contains the inventory of the neighbouring nodes [10]. This approach is based on a push-update technique where each peer sends to its neighbouring nodes information about its resources and constantly updates them whenever its resources change. Similar approaches are exploited in the *Directed Breadth-First Search* [8] and in the *Intelligent Search* mechanism [11] where each peer in the network builds a profile of its peers and uses the profile to determine which peers are more likely to answer each query.

A self-learning approach is the basis of the *Adaptive Probabilistic Search* [12], where each peer uses feedback from previous searches to adjust the probability of successfully using certain neighbouring peers in future searches. This approach constitutes an advantage over the ones proposed in [8], [9] and [11] because it does not introduce an excessive overhead to update the indexes at the neighbours. A more flexible feedback-based approach is employed by the *Directed Searches* [13], where peers use a vast set of metrics, which range from the number of successfully returned query results to network connectivity and latency, to learn from previous interactions and improve future searches.

B. Search Mechanisms in Structured Networks

One approach used to improve the *uninformed search* mechanisms in unstructured *P2P* networks, described in

the previous sub-section, was based on the use of indexes and statistical information to help peers choose the appropriate neighbours to which future search queries should be routed. Another approach is to introduce some sort of structure to improve message routing, which is usually done by partitioning the network into a set of communicating clusters of peers that are connected amongst them by a network of *super-peers* [7]. A *super-peer* belongs to a higher-level of a peer's hierarchy, which is usually based on content-related criteria. Super-peers are responsible for managing and facilitating search processes among the peers in its cluster (by maintaining an index of its peers' resources) and for communicating with neighbouring *super-peers* to further extend search processes that could not be resolved *locally*. An example of this structured approach was introduced in the *FastTrack P2P* platform [14] to handle the scaling problems of the Gnutella protocol.

Hierarchical approaches such as the ones based on these special-purpose peers come at the expense of resilience to semi-catastrophic failures of *super-peers* near the top of the hierarchy [15]. In order to offer a scalable and yet robust infrastructure for *P2P* networks, an alternative approach, based on the *Distributed Hash Tables* (*DHT*) abstraction, has been proposed. *Chord* [16], *Pastry* [17] and *Tapestry* [18] are examples of DHT implementations. The *Content-Addressable Network* (*CAN*) [19] differs from these approaches by operating in a multi-dimensional view of the *DHTs*, i.e. by allowing for peers to search for resources in the network using more than one type of key simultaneously. The *DHT* approach is based on the sole principle that a resource can be identified by a numeric key that is created through a *hash* function, based on the resource's contents. In order for a resource to be published under a specific key, the peer routes the publishing request to the peer with the key closest to the resource's key (based on some "*closeness*" function), which in turn store that information in a routing table. When a peer searches a specific resource in the network, it routes the request to the peer with the closest key, which in turn will apply the same process until the resource is located in the network.

III. RESOURCE COORDINATION IN P2P NETWORKS

At an abstract level, a resource can be viewed as a computational capability that is offered by a certain entity. At a more concrete level, a resource can be instantiated as a web service, a file, an intelligent agent capability, storage or processing capabilities or any other computational skill available in a network of interconnected peers. In distributed networks, where the goal is to build a collaborative environment to facilitate resource sharing, resources need to be easily located in order to be composed and executed. Resource coordination research addresses these issues and aims at creating an environment where peers, managing different resources, can cooperate to provide value-added services, which could not be provided if the peers were to operate individually. In *P2P* networks, this assumes even greater importance as the diverse and distributed environment

offers a potential for building powerful applications based on resource coordination.

Resource discovery constitutes the first and most important step on every collaborative environment and early attempts for resource discovery in large networks [20] based the process on the use of dedicated central servers. However, centralized solutions were deemed unsuitable for large environments and later approaches decided to use the hybrid potential of *P2P* networks, such as dynamic federated environments [21], where super-peers share their peers' resources by federating with other content-related super-peers; structured networks with resource rating [22]; and distributed multi-registry centres [23][24], where peers register their resources in the appropriate registry centres based on their type or domain in which they operate. Unfortunately, without some concrete way to describe relationships between resources, these approaches do not leverage the potential of semantically-linked peers to improve the resource coordination process. Semantic links aim at providing a more meaningful way to connect peers and their resources, thus allowing for peers to easily combine their resources with other semantically-related peers.

In the following sub-sections, we describe some approaches that use semantic-links to connect peers in order to provide an optimized resource discovery environment. Semantic links between peers are based on the properties of their resources. We also present some multi-agent based approaches that address the resource coordination problem by presenting concrete discovery, composition and execution mechanisms.

A. Semantic-Link Based Resource Coordination

An effective way to optimize the resource discovery process is to establish semantic connections between peers based on the properties of their resources. If, for example, a peer manages a resource which is somehow related to another resource that is managed by another peer, then it is important that a semantic-based connection exists between these two peers stating the meaning of their relationship. This semantic-link can then be used to improve future searches or collaboration initiatives. An area which has explored the power of semantic descriptions is the *Semantic Web* [25], a world-wide initiative to bring semantic meaning to the realm of web services. *Semantic Web Services* are usually provided by peer-based internet end-points that, all together, build a large-scale network of distributed resources. Using *P2P* computing and semantic descriptions of web services, several research approaches have addressed resource coordination issues especially related to resource matchmaking, discovery and composition.

A decentralized web service organization approach is presented in [26], in which a *DHT*-based catalog service is used to store the semantic indexes for direct service publication and discovery. This semantic indexation consists of a classification of the services based on domain-related categories. A similar approach was described in [27], where peers in a network advertise

their "*service expertise*" based on domain categories. The algorithm used to spread the advertisements within the *P2P* network is based on a ranking system, which allows peers to route their "*service expertise*" only to peers that operate in similar domain categories (according to a *similarity* function).

GloServ [28] uses a keyword-based taxonomy search on a hierarchical hybrid *P2P* network to build a semantic overlay between the peers that operate in the same (sub) domain. Several other keyword-based mechanisms for semantic web services discovery and matchmaking on *P2P* networks that do not rely on centralized taxonomies or domain categories were proposed. The keyword search in these approaches is done at the level of operation names [29] or non-functional service descriptions [30], [31]. Even though these attempts to create semantic links between peers and resources may help improve resource discovery, the established relationships can be further improved by using more meaningful properties.

The *Web Services Peer-to-Peer Discovery Service* (*WSPDS*) [32] is a service discovery approach in pure *P2P* networks, where semantic links between peers are based on the similarity of the services they provide. The matchmaking process is done in a deductive-based service profile matching based on the comparison of the resource's *inputs* and *outputs*. A similar matchmaking process is suggested in *Bibster* [33], where peers' capabilities are semantically-linked by first applying the same deductive-based *inputs* and *outputs* comparison as in WSPDS and then by ranking services through a similarity-based expertise matching. The *METEOR-S Web Service Discovery Infrastructure* [34] presents a similar approach to the *WSPDS* and *Bibster* but it relies on a hybrid *P2P* network architecture where special peers are introduced to handle a global ontology. The approach presented in [35] also uses semantic matching at the level of *inputs* and *outputs* but it differs from related approaches by using a *DHT*-based service discovery process on top of a *Chord P2P* network.

B. Multi-Agent Based Resource Coordination

The use of Multi-Agent Systems (*MAS*) to efficiently coordinate resources in collaborative environments has gained a lot of attention, especially due to the advances in *P2P* computing. The evolution of search mechanisms, which were showing signs of scalability and robustness, opened the road for the development of more complex and intelligent systems.

Some systems rely on structured solutions, such as aggregation of peers in communities or the use of middle layers that have specific coordination capabilities. *SELF-SERV* [36] is a framework where web services are declaratively composed based on state charts and the resulting composite services are executed in a decentralized way within the *P2P* dynamic environment. This framework relies on the concept of service communities (containers of alternative services), which provide abstract descriptions of desired services and allow actual service providers to register in the appropriate community. The distributed execution is managed by *coordinator* agents, which are in charge of

initiating, controlling, monitoring and collaborating with their peers to manage the execution of the services which they are attached to.

The approach presented in [3] uses a *MAS* to perform distributed composition of web services, based on agents that play the role of mediators. In [37] a similar approach to [3] is used for automated web service composition over a *P2P* network, where peers are organized into communities that represent the same domain. The major difference between this approach and [3] is that the former tries to determine links between web services at publishing time (suitable for more stable networks) and the latter does this at composition time (suitable for more dynamic networks).

A-peer [38] is a multi-agent-based *P2P* system where agents rely on hierarchically arranged advertising elements to find the services they need from other agents. This kind of middleware solution is also used in [39], which describes a framework for agent-enabled web service composition where an *Agent Middle Layer* is used to transform service requests into the corresponding tasks in the *P2P* environment.

Structured systems contribute to optimize the routing mechanisms in *P2P* computing, however, at the cost of introducing central points of failure and in certain environments, compromising scalability. To avoid these failure-prone solutions, some approaches are based on pure *P2P* networks. An inference system based on pure *P2P* networks is presented in [40]. In this approach, each peer can answer queries by reasoning from its local (propositional) theory but can also perform queries to some other peers with which it is semantically related by sharing part of its vocabulary. In order to create these semantic relations (referred by the authors as *acquaintance networks*), new peers joining the *P2P* system simply declare their *acquaintances* in the network, i.e., the peers it knows to be sharing variables with, and it declares the corresponding shared variables. However, the authors do not clearly explain how this "*acquaintances declaration*" process is carried out efficiently in the *P2P* network.

The study of ant communities has inspired some research on the development of *P2P* systems based on multi-agent systems. *Anthill* [41] is a *P2P*-based *MAS* which emulates the resource coordination behaviour of ants. In this framework, storage or computational resources (referred to as "*nests*") generate requests (referred to as "*ants*") in response to user requests. These *ants* travel across the network of *nests* in order to be processed and executed. *Ants* do not communicate directly with each other. Instead, they communicate indirectly by leaving information related to the service they are implementing in the appropriate resource manager found in the visited *nests*. This *pheromone*-like approach, also called "*stigmergy*", allows the network to self-organize and improve its performance over time. The idea of assigning agents to carry on requests (*ants*) avoids a non-scalable flooding search technique, since each *ant* will only travel to a *nest* at a time and it will not replicate. However, the search performance might be slower

because each edge of the network (*nests*) is only travelled once at a time for each request. The selection of the next *nest* to be visited by an *ant* can either have a deterministic approach (once the network is organized and appropriate overlay networks are available) or a totally random (*uninformed*) approach. A similar approach to [41] is proposed in [42], where mobile agents use *pheromone*-like behaviour to optimize the trails within a *P2P* network. However, instead of using the update process based on the discovered path, as in [41], the mobile agent creates a referral to the query-answering node, thus creating a direct link that will improve future similar searches.

A fully distributed approach to the resource discovery problem in a multi-agent system is presented in [43]. In this system, each agent maintains a limited size local cache in which it keeps information about different resources and the agents that provide them. An agent searching for a specific resource contacts its local cache and if there is no information for the resource, it contacts a *k*-random subset of neighbours (to avoid *flooding*), which in turn contact their neighbours. The process goes on until the resource is found in some cache. Also, this system innovates from similar search mechanisms approaches ([8], [11] and [45]) by proposing the use of *inverted caches*. Besides maintaining a local cache of agents with certain resources, the agent maintains a cache of agents that have a reference to its own resource in their caches to facilitate the mechanism of updating changes in the network. However, this approach does not address the problem of choosing the appropriate resources that each agent should maintain in its cache. Doing so could help improve search performance in the network over time.

IV. IMPROVING RESOURCE COORDINATION IN *P2P* NETWORKS

The use of semantic-links can be useful when it is used to optimize the coordination between the entities that manage resources in some collaborative environment. This optimization can be achieved by using knowledge acquired in previous interactions to improve future interactions. However, the recent approaches for resource coordination in P2P networks, described in the previous section, still lack a sense of constant network evolution based on the self-organization process. These approaches dynamically self-organize but then stop this process once the necessary structure is obtained. The constant dynamic adaptation, network evolution and self-organization assume very important roles in the development of more robust and scalable intelligent dynamic environments. In this section we present a set of proposals for improving the resource coordination process in *P2P* networks by using a powerful *semantic overlay network* [45] that is dynamically built by the search mechanism itself.

The discovery process, in which peers establish semantic connections with other peers thus fuelling the *semantic overlay network*, is first carried out by using specific search mechanisms. Our approach, with which we intend to improve the discovery process performance, is described in sub-section A of this section. Once the

semantic overlay network is built, besides using it to easily locate resources, peers can use it to perform more complex tasks such as composition of resources in order to achieve more ambitious goals. The creation and use of the *semantic overlay network* is described in sub-section B of this section.

A. Efficient Search Mechanisms

When a *P2P* network is first established, peers are not aware of other peers' resources and usually only have meaningless structural connections to a set of neighbouring peers. In order to create a useful network with peers or resources with which it has some semantic relation or similarity, a peer must discover them within this network of (yet meaningless) connections. As previously described in section II, a lot of search mechanisms have been proposed, which range from *flooding* and *random walks* techniques to *informed searches* and *distributed hash tables*. In this sub-section we present two different proposals to efficiently search peers and resources in a *P2P* network. Both approaches assume that it is possible to univocally identify search queries to avoid having peers processing the same query twice. We also present some network evolution techniques to improve the performance of the search mechanisms throughout time.

Priority-based Flooding

This approach is based on the assumption that a *flooding* technique is only inefficient if the network is already overloaded with requests. If peers are idle, then the *flooding* mechanism is, in fact, the fastest and most complete way of delegating a search query. However, it is difficult for a peer to determine whether or not its neighbours have a heavy work-load at a certain moment.

We introduce the concept of *priority-based flooding*, which allows peers to assign a priority to search queries based on their propagation level within the network. The principle of this search mechanism is very simple: peers use the propagation level of a search to inversely calculate the priority of the query, i.e., the highest the propagation level, the lower the priority. The following algorithm may be used for the search mechanism: let Q be the list of queries currently waiting to be processed (peer's work load), N the list of neighbours, mq and mp auxiliary variables indicating the maximum priority query and its priority, p the priority of a request and r the result of a query processing event.

Algorithm: Priority-based Flooding
PbF(Q, N)
(1) $mp = \varnothing$
(2) $mq = \varnothing$
(3)   **foreach** $q_i \in Q$
(4) $p = 1/depth(q_i)$
(5)   **if** ($p > mp$)
(6) $mp = p$
(7) $mq = q_i$
(8)   **end if**
(9)   **end foreach**

(10) $r = process(mq)$ // *mq is the highest priority query*
(11) **if** $(r = \emptyset)$
(12) $depth(mq) = depth(mq) + 1$
(13)   **foreach** $n_i \in N$
(14) $reply(forward(mq, n_i))$
(15)   **end foreach**
(16) **else**
(17) $reply(r)$
(18) **end if**

Using this algorithm, peers can efficiently manage their work load by giving priority to *local* requests (search queries triggered by close peers) in detriment of requests originated by far-away peers. We believe that this is a fair policy for peers to use, since it relies on the fact that if the propagation level of a search query is high, then the number of nodes that have had access to the search query is also quite high. Hence, the probability for the search query to be processed by some other peer with a lower work load is also high. This approach may allow to increase (or maybe even eliminate) the *Time-To-Live* property of *flooding* search queries.

Iterative Branching Depth-first Search

This search technique is based on the *depth-first* search mechanism and it can be used as an alternative to the *priority-based flooding* (in high load networks). We introduce the use of an iterative process in the *depth-first* search to increase the coverage of the network. When initiating a search query, a peer will randomly contact one of its neighbours. If the neighbour immediately replies with the answer, then the process ends. If the neighbour replies stating that it does not have the answer and that it will apply the same *iterative branching depth-first* search process with its neighbours, then the peer will contact a second neighbour and so forth. The following recursive algorithm depicts the steps of the search mechanism: let q be the query to be processed, N the list of neighbours and r the result of a query processing event.

Algorithm: Iterative Branching Depth-First Search

IBDFS(q, N)
(1) $r = \emptyset$
(2)   **if** $(\neg\, processed(q))$
(3) $r = process(q)$
(4) $reply(r)$
(5) $processed(q) = $ true;
(6)   **end if**
(7)   **if** $(r = \emptyset)$ // *else terminate execution*
(8)   *randomly select* $n_i \in N$
(9)   **if** $(n_i \neq \emptyset)$ // *else terminate execution*
(10) $rn_i = forward(q, n_i)$
(11)   **if** $(rn_i = \emptyset)$ // *n_i will apply the same process*
(12)   IBDFS$(q, N - n_i)$
(13)   **else**
(14) $reply(rn_i)$
(15)   **end if**
(16)   **end if**

---

(17) **end if**

This approach increases the branching level iteratively on each hop count, thus increasing the chances of finding the answer faster, comparatively to the *depth-first* search approach. Fig. 1 presents a comparison of the number of peers reached (darker nodes) in a search query with the same hop count. This algorithm can also be adapted to consider *k* number of neighbours (instead of just one), in each iteration.

Network Evolution Techniques

In the beginning of the discovery process, peers in a *P2P* network do not have enough information about other peers. As they go along, the interactions between the peers are valuable sources of information that can be used to improve the performance of future searches. Furthermore, the use of informed search techniques scales a lot better throughout time as peers improve their connections with other peers based on previous interactions [44].

In order to improve the performance of the proposed search mechanisms, we propose some adaptation procedures for the peers which we believe will improve the searches throughout time. These procedures contribute to the evolution of the network by denoting some self-organization that will improve future searches (by reducing the query response time and the network bandwidth usage and maximizing the accuracy of the results) through the dynamic creation of a semantic overlay network (see Fig. 2).

In order for a peer to improve its participation in future searches, it is important that it caches previous search contributions. For example, as a query response travels back to the requester node, all nodes in that specific path can either store the response themselves or cache a link to the node which has the response, thus working as a referral for future searches. However, after some time contributing to search queries, it may happen that peers hold a huge cache of referrals that becomes intractable as they contribute more and more throughout time. In order

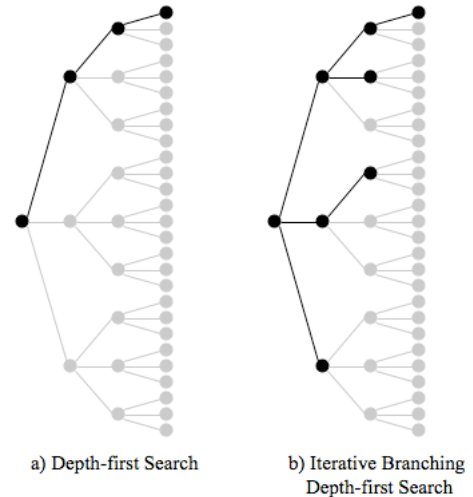a) Depth-first Search   b) Iterative Branching Depth-first Search

Figure 1. Comparison between (a) *Depth-first Search* and (b) *Iterative Branching Depth-first Search* in a search with the same hop count of 3.
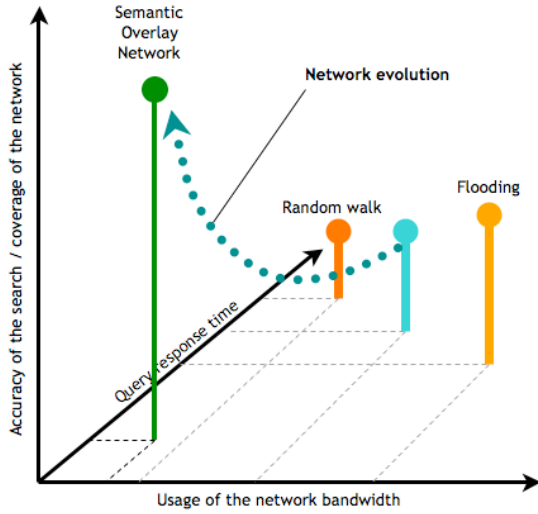
Figure 2. Network evolution[1]

Figure 3. Creation of a *Semantic Overlay Network*

to avoid loss in performance due to the size of the cache, peers can store only a fixed number of references and decide which ones to store based on a metric, such as the frequency of the request. Even though rare requests have lower performance in this process, frequent search queries will be optimized, which globally seems to be a good assumption.

An alternative approach can be based on a direct link between the responder node and the requester node. If we consider the *priority-based flooding* algorithm, we see that this causes a massive generation of reply messages (lines (11) and (12) of the *PbF* algorithm). To avoid this situation, we can change the search mechanism so that the query response is returned directly to the query requester, instead of being carried back through the original path. For example, if a peer A has the response for the query made by a peer B, A will directly send the response to B. Even though the peers on the recurring path will not learn the result of the query, the result will reach the requester node faster and a lot of messages can be saved. Furthermore, the nodes that participated in the search, even if just for forwarding or propagating the request, can assume that, after some time, the requester node has already received the necessary response. Hence, future similar searches (for example, a peer C requesting the same contents as B) can be referred to the previous requester node (B), which in turn can refer it to the responder node (A) or provide itself the response directly (to C).

B. Semantic Overlay Network and Semantic-based Resource Coordination

Although semantic-free approaches (such as *DHT*) provide good performance for point queries (where the search key is known exactly), they are not as effective for approximate, range, or text queries [45] and they do not, on their own, capture the relationships between the
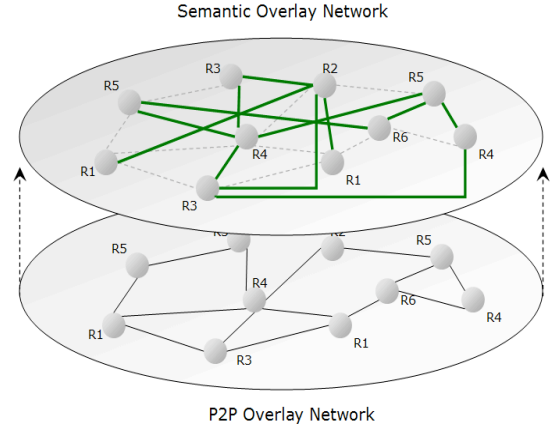
resource or peer's name and its content or metadata [46]. A semantic approach aims at bringing a more powerful and meaningful description of peers and their resources so as to provide a way to easily establish fruitful relationships between them. A *semantic overlay network* [45] represents the interconnections of semantically related nodes.

In our approach, as depicted in Fig. 3, we use a *semantic overlay network* to establish relationships between peers in order to facilitate the coordination of the resources that they manage. These relationships are based on semantic dependencies between resources. To this end, we view resources from a semantic point of view. Semantic resources can have inputs, outputs, pre-conditions (conditions which have to be true in order for the service to be executed) and effects (conditions that become true after the execution of the service). These properties describe resources in a more meaningful way supporting better matchmaking, discovery and composition processes.

Our approach is a two-folded resource coordination process. In the first process, referred to as *self-organization* process, a peer uses the previously described search mechanisms (see section IV.A) to find resources that depend on its own resources and vice-versa. During this process, a peer uses a simple inference rule to determine whether or not other peers' resources should be semantically linked to its own resources. For example, consider a peer A with resource X and a peer B with resource Y. A's resource X should be semantically linked to B's resource Y if Y's effects contribute to achieve X's pre-conditions, as illustrated in the following expression[2]:

$$\exists\, c\, [\, (c \in preconditions(X)) \wedge (effects(Y) \vdash c)].\quad (1)$$

The main purpose of this stage is to allow a network of otherwise unrelated peers to self-organize, such that each peer knows exactly where the resources on which it depends are.

While this network of semantic interconnections is

---

being created, the second process, referred to as *coordination* process, where the self-organized network is able to receive requests to solve specific problems, can take place. The request, which describes an *initial state* and a set of *goals*, is sent to a peer that will process them and determine whether or not it can contribute to solve the problem. This analysis is carried out by the same inference rule as in the first process, where the peer will determine if, for example, its resource Y's effects contribute to achieve the goals of problem P:

$$\exists\, c\, [\, (c \in goals(P)) \wedge (effects(Y) \vdash c)]. \qquad (2)$$

If the peer is not able to contribute to the solution of the problem, it uses its acquired knowledge to find the appropriate peer to forward the request to. Once the appropriate peer (or set of peers) is found (peers whose resource's effects contribute to solve the goals of the problem), the composition of the resources is carried out instantaneously (if all the necessary semantic interconnections have already been established). This happens because the *semantic overlay network* (created in the previously described process) specifies the dependencies between the peers. The process halts once a resource or set of resources is found whose pre-conditions are satisfied by the *initial state* described in the problem. The problem is finally solved by the execution of the created composition plan that describes which resources should be used and how they should be combined.

V. SIMULATION RESULTS AND EVALUATION

In order to assert if the proposed search algorithms have better performance than existing approaches, we have performed several simulations regarding the generation of a semantic overlay network from a network of randomly-connected peers. These simulations are still preliminary and the results are not yet conclusive. We intend to perform more complete simulations by testing different configurations of the environment, namely by analysing if the number of each agent's neighbors, the total number of agents and the network topology (totally random networks, small-world networks) have an impact on the performance of the algorithms. For these simulations, we tested an environment of 1000 agents all with different skills and randomly connected to 3 neighbors each. We use the term *Network Completeness* to define the number of agents that have reached their status of connectivity in the semantic overlay network, that is, they have found the agents to which they are semantically related.

Comparatively to the *Depth-First Search* (*DFS*) algorithm, the *Iterated Branching Depth-First Search* (*IBDFS*) algorithm has proven to be a much faster alternative, as Fig. 4 shows. This improvement is a result of the branching factor of the algorithm, which increases the parallel power of the search, by iteratively distributing the query to the agent's neighbors while the desired result is not obtained.
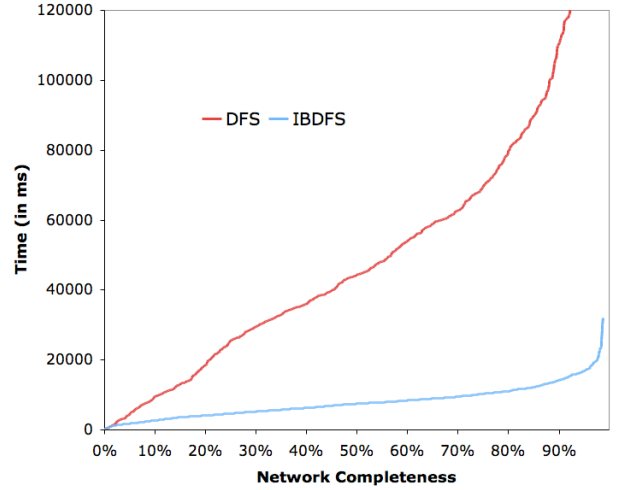

Figure 4. Comparison between DFS and IBDFS

We have also compared the *IBDFS* algorithm with the *Flooding* algorithm and the *Priority-based Flooding* (*PbF*) algorithm. Fig. 5 presents the result of those simulations. As depicted in the figure, the *IBDFS* algorithm is the fastest to reach an almost complete (90%) semantic overlay network, while the Flooding and the PbF algorithms have a very similar behavior. However, IBDFS seems to loose some performance to the other algorithms towards the total completeness of the network.

VI. CONCLUSIONS AND FUTURE WORK

We have presented a survey on multi-agent based resource coordination in *P2P* networks and have further described our approach for creating an innovative process for coordinating agents and resources in unstructured distributed networks. Our approach is based on a *semantic overlay network*, which is created by employing sophisticated efficient search mechanisms that dynamically create and take advantage of a network of semantic dependencies between peers and their resources.

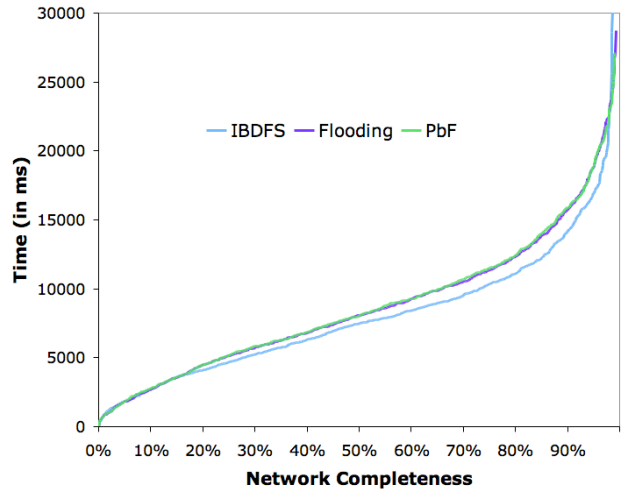Network evolution and self-organization properties are

Figure 5. Comparison between IBDFS, Flooding and PbF

present in our proposal to reflect the necessary adaptation to the typical dynamics of *P2P* networks. However, careful attention needs to be taken to ensure a balance between the overhead created by the self-organization stage and the rate at which peers join or leave the network. We intend to further analyse this issue, by determining if this interleaved two-process approach is suitable for highly dynamic networks.

Our future work will focus on performing more complete tests (as the ones presented in section V are still very preliminary) with the described search mechanisms and compare the results with related work and also on the implementation, test and analysis of the semantic-based resource coordination infrastructure.

REFERENCES

[1] Napster, LLC. "Napster", *http://www.napster.com*, 2003.
[2] Gnutella Developer Forum. "Gnutella protocol specification v0.4," *http://rfc-gnutella.sourceforge.net/developer/stable/index.html*, 2002.
[3] P. Küngas, and M. Matskin, "Semantic web service composition through a p2p-based multi-agent environment," *Springer-Verlag, Lecture Notes on Computer Science,* vol. 4118, pp. 106-119, 2006.
[4] S. Willmott, J. M. Pujol, and U. Cortes Garcia. "On exploiting agent technology in the design of peer-to-peer applications," *Springer-Verlag Lecture Notes on Computer Science*, vol. 2872, 2005.
[5] H. Helin, M. Klusch, A. L. Lopes, A. Fernandez, M. Schumacher, H. Schuldt, F. Bergenti, and A. Kinnunen, "CASCOM: Context-aware service co-ordination in mobile p2p environments," *Springer-Verlag, Lecture Notes on Computer Science*, vol. 3550, pp. 242-243, 2005.
[6] A. L. Lopes, and L. M. Botelho, "Task decomposition and delegation algorithms for coordinating unstructured multi agent systems," In *Proc. of the Int. Conf. on Complex, Intelligent and Software Intensive Systems,* pp. 209-214, 2007.
[7] D. Bianchini, V. De Antonellis, M. Melchiori, and D. Salvi, "Peer-to-peer semantic-based web service discovery: state of the art, " *Technical Report, Dipartimento di Elettronica per l'Automazione Università di Brescia*, 2006.
[8] B. Yang, and H. Garcia-Molina, "Efficient search in peer-to-peer networks," In *Proc. of Int. Conf. on Distributed Computing Systems,* 2002.
[9] A. Crespo, and H. Garcia-Molina, "Routing indices for peer-to-peer systems". In *Proc of the 22nd Int. Conf. on Distributed Computing Systems*, pp. 23-30, 2002.
[10] O. Ratsimor, D. Chakraborty, A. Joshi, T. Finin, and Y. Yesha, "Service discovery in agent-based pervasive computing environments," *Mobile Network Applications*, 9(6), pp. 679-692, 2004.
[11] V. Kalogeraki, D. Gunopulos, and D. Zeinalipour-Yazti, "A local search mechanism for peer-to-peer networks," In *Proc. of the 11th Int. Conf. on Information and Knowledge Management*, pp. 300-307, 2002.
[12] Q. Tsoumakos, and N. Roussopoulos, "Adaptive probabilistic search for peer-to-peer networks," In *Proc. of the 3rd Int. Conf. on Peer-to-Peer Computing,* pp. 102-110, 2003.
[13] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and replication in unstructured peer-to-peer networks". In *Proc. of the 16th Int. Conf. on Supercomputing*, ACM Press, New York, NY, pp. 84-95, 2002.
[14] J. Liang, R. Kumar, and K. W. Ross, "The FastTrack overlay: a measurement study," *Computer Networks,* 50(6):842-858, 2005.
[15] H. Balakrishnan, M. F. Kaashoek, D. Karger, R. Morris, and I. Stoica, "Looking up data in p2p systems," *Communications of the ACM,* Vol. 46, Nº. 2, pp. 43-48, 2003.
[16] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," In *Proc. of the ACM SIGCOMM 01 Conference*, 2001.
[17] A. I. Rowstron, and P. Druschel, "Pastry: scalable, decentralized object location, and routing for large-scale peer-to-peer systems," In *Proc. of the IFIP/ACM Int. Conf. on Distributed Systems Platforms Heidelberg*, *Springer-Verlag, Lecture Notes On Computer Science*, vol. 2218. pp. 329-350, 2001.
[18] B. Y. Zhao, J. D. Kubiatowicz, and A. D. Joseph, "Tapestry: an infrastructure for fault-tolerant wide-area location and routing," *Technical Report. UMI Order Number: CSD-01-1141., University of California at Berkeley*, 2001.
[19] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," In *Proc. of ACM SIGCOMM'01 Conference*, 2001.
[20] M. A. Sheldon, A. Duda, R. Weiss, and D. K. Gifford, "Discover: a resource discovery system based on content routing," In *Proc. of the 3rd Int. World-Wide Web Conference on Technology, Tools and Applications*, pp. 953-972, 1995.
[21] K. Sivashanmugam, K. Verma, and A. Sheth. "Discovery of web services in a federated registry environment," In *Proc. of the IEEE Int. Conf. on Web Services*, pp. 270-278, 2004.
[22] F. Emekci, O. D. Sahin, D. Agrawal, A. El Abbadi. "A peer-to-peer framework for Web service discovery with ranking," In *Proc. of the IEEE Int. Conf. on Web Services*, pp. 192- 199, 2004.
[23] Q. Lin, R. Rao, M. Li. "DWSDM: a web services discovery mechanism based on a distributed hash table," In *Proc. of the 5th Int. Conf. on Grid and Cooperative Computing Workshops*, pp. 176-180, 2006.
[24] T. Gagnes, T. Plagemann, and E. Munthe-Kaas, "A conceptual service discovery architecture for semantic web services in dynamic environments," In *Proc. of the 22nd Int. Conf. on Data Engineering Workshops,* IEEE Computer Society, Washington, DC, p. 74, 2006.
[25] T. Berners-Lee, J. Hendler, and O. Lassila. "The semantic web," *Scientific American*, 279, 2001.
[26] S. Yu, J. Liu, J. Le, "Decentralized web service organization combining semantic web and peer to peer computing," *Springer-Verlag, Lecture Notes on Computer Science*, vol. 3250, pp. 116-127, 2004.
[27] H. Jin, H. Wu, Y. Li, and H. Chen, "Semantic overly-driven web services discovery," In *Proc. of the 1st Int. Conf. on Semantics, Knowledge and Grid*, pp. 76-87, 2005.

[28] K. Arabshianand, and H. Schulzrinne, "An ontology-based hierarchical peer-to-peer global service discovery system," In *Journal of Ubiquitous Computing and Intelligence*, 2006.

[29] J. Liu, and H. Zhuge, "A semantic-link-based infrastructure for web service discovery in P2P networks, " In *Special interest Tracks and Posters of the 14th Int. Conf. on World Wide Web.* ACM Press, New York, NY, pp. 940-941, 2005.

[30] I. Toma, B. Sapkota, J. Secuila, J. M. Gomez, D. Roman, and C. Bussler, "P2p discovery mechanisms for web service execution environment," In *Proc. of the 2nd WSMO Implementation Workshop*, 2005.

[31] B. Sapkota, L. Vasiliu, I. Toma, D. Roman, and C. Bussler, "Peer-to-peer technology usage in web service discovery and matchmaking," In *Springer-Verlag, Lecture Notes on Computer Science*, vol. 3806, pp. 418-425, 2005.

[32] F. Banaei-Kashani, C. Chen, and C. Shahabi, "WSPDS: web services peer-to-peer discovery service," In *Proc. of the Inter. Conf. on Internet Computing*, pp. 733–743, 2004.

[33] P. Haase, S. Agarwal, and Y. Sure, "Service-oriented semantic peer-to-peer systems," In *Proc. Of the WISE Workshop on Intelligent Networked and Mobile Systems*, pp. 46-57, 2004.

[34] K. Verma, K. Sivashanmugam, A. Sheth, A. Patil, S. Oundhakar, and J. Miller, "METEOR-S WSDI: a scalable infrastructure of registries for semantic publication and discovery of web services," In *Journal of Information Technology and Management, Special Issue on Universal Global Integration*, 6(1):17–39, 2005.

[35] R. Romeikat, and B. Bauer, "Towards semantically-enhanced distributed service discovery," In *Proc. of the 2nd Int. Conf. on Internet and Web Applications*, 2007.

[36] B. Benatallah, Q. Z. Sheng, A. H. H. Ngu, and M. Dumas, "Declarative composition and peer-to-peer provisioning of dynamic web services," In Proc. of the 18th Int. Conf. on Data Engineering, p. 297, 2002.

[37] I. B. Arpinar, B. Aleman-Meza, R. Zhang, and A. Maduko, "Ontology-driven web services composition platform," In *Proc. of the IEEE Int. Conf. on E-Commerce Technology.* IEEE Computer Society, Washington D.C., pp. 146-152, 2004.

[38] T.-Y. Li, Z.-G. Zhao, and S.-Z. You, "A-peer: an agent platform integrating peer-to-peer network," In *Proc. of 3rd IEEE/ACM Int. Symposium on Cluster Computing and the Grid*, pp. 614- 617, 2003.

[39] V. Ermolayev, N. Keberle, S. Plaksin, V. Terziyan, O. Kononenko, "Towards a framework for agent-enabled semantic web service composition," In *International Journal of Web Service Research*, vol. X, 2004.

[40] P. Adjiman, P. Chatalic, F. Goasdoué, M.-C. Rousset, and S. Laurent, "Distributed reasoning in a peer-to-peer setting: application to the semantic web," In *Journal of Artificial Intelligence Research*, 25, pp. 269-314, 2006.

[41] O. Babaoglu, H. Meling, and A. Montresor, "Anthill: a framework for the development of agent-based peer-to-peer systems," In *Proc. IEEE ICDCS'02*, pp. 15-22, 2002.

[42] P. Dasgupta, "Improving peer-to-peer resource discovery using mobile agent based referrals," In *Proc. of the Int. Workshop on Agents and Peer-to-Peer Computing, Springer-Verlag, Lecture Notes on Computer Science*, vol. 2872, pp. 186-197, 2004

[43] V. V. Dimakopoulos, and E. Pitoura, "A peer-to-peer approach to resource discovery in multi-agent systems," In *Proc. of CIA2003, Springer-Verlag, Lecture Notes on Computer Science,* vol. 2782, pp. 62-77, 2003.

[44] G. H. L. Fletcher, H. A. Sheth, K. Börner, "Unstructured peer-to-peer networks: topological properties and search performance," In *Proc. of the Int. Workshop on Agents and Peer-to-Peer Computing, Springer-Verlag, Lecture Notes on Computer Science*, vol. 3601, pp. 14-27, 2004.

[45] A. Crespo, A. and H. Garcia-Molina, "Semantic overlay networks for p2p systems," In *Proc. of the Int. Workshop on Agents and Peer-to-Peer Computing, Springer-Verlag, Lecture Notes on Computer Science*, vol. 3601, pp. 1-13, 2005.

[46] J. Risson, T. and Moors, "Survey of research towards robust peer-to-peer networks: search methods," *Comput. Networks* 50(17):3485-3521, 2006.

António L. Lopes was born in Cascais, Portugal on February, 18th of 1979. He has obtained a degree on Computer Science and Enterprise Management in 2002 and a M.Sc. on Computer Science and Telecommunications Engineering in 2005, in ISCTE, a public university in Lisbon (Portugal) for his research on context-aware execution of semantic web services. The author is currently a Ph.D. student of Artificial Intelligence in ISCTE in the area of resource coordination in multi-agent based p2p networks.

He has been a Researcher in the *We, the Body, and the Mind* Research Lab of Adetti-ISCTE since January 2000 and, in certain occasions, an Assistant Lecturer in computer science under-graduation courses in the university. His current research interests are related to his Ph.D. in the area of coordination of resources that are spread across unstructured networks of intelligent agents.

Luís M. Botelho was born in Lisbon, Portugal on December, 18th of 1959. He has obtained an Electrotechnical Engineering degree in 1984 and M.Sc. in Electrotechnical Engineering and Computers in 1989, both by IST (Instituto Superior Técnico), Technical University of Lisbon, Lisbon, Portugal. He has finished his Ph.D. in 1997, with thesis in Artificial Intelligence, by ISCTE (Instituto Superior de Ciências do Trabalho e da Empresa), Lisbon, Portugal. The author's major field of study is artificial intelligence.

He is an Associate Professor of the Department of Information Sciences and Technologies of ISCTE and a Senior Researcher of Adetti, a private research center associated with ISCTE in Lisbon, Portugal. He is author and co-author of several international journal, conference and workshop papers on artificial intelligence. Current research interests include agent-based service coordination, and the role of the body of artificial agents on their behaviour, mainly on symbol grounding.

Prof. Luis M. Botelho is a member of the Portuguese Association of Artificial Intelligence and was the representative of Adetti in FIPA (Foundation for Intelligent Physical Agents), a standardization body on agent technology.